

69/876,993

Set	Items	Description
S1	1013084	ROW OR ROWS OR COLUMN? OR TUPLE? OR ATTRIBUTE? OR FIELD?
S2	3319219	REPOPULAT? OR RELOAD? OR REPLAC? OR REFILL? OR SWITCH? OR - MODIF? OR CHANGE? OR ALTER?
S3	3210200	FILL OR WRITE OR FILLING OR FILLS OR WRITING OR LOAD? OR - POPULATING OR POPULATES OR SET
S4	4070471	VALUE? OR DATA? OR RECORD? OR CELL OR CELLS
S5	190560	EMPTY OR UNFILLED OR UNUSED OR BLANK OR BLANKS OR UNPOPULA- TED OR CLEARED OR VACANT
S6	652036	OLD OR PRIOR OR PREVIOUS OR LAST OR OLDER OR FORMER OR PAST OR PRECEDING OR EARLIER
S7	36	S1 AND S2 AND S3 AND S4 AND S5 AND S6
S8	36	IDPAT (sorted in duplicate/non-duplicate order)
S9	36	IDPAT (primary/non-duplicate records only)
S10	13	S9 AND IC=G06F
S11	10	S10 NOT AD=20010714:20030714
S12	9	S11 NOT AD=20030714:20050701
S13	389421	DATABASE? OR DATABANK? OR DATA() (BASE? OR BANK?) OR DB OR - OODB OR RDB OR DBMS OR RDBMS OR ARRAY
S14	653	S13 AND S2 AND S6(2N)S4
S15	202493	S1 AND S2
S16	113	S14 AND S15
S17	50	S16 AND IC=G06F
S18	42	S17 NOT AD=20010704:20030704
S19	41	S18 NOT AD=20030704:20050611
S20	39112	S1(3N)S2
S21	9	S19 AND S20
S22	4	S1(2N) (DELETED OR CLEARED OR EMPTIED OR S5) AND (S2(2N)S6)
S23	128913	S2(2N)S4
S24	46763	S6(3N) (S1 OR S4)
S25	3303	S23 AND S24
S26	42004	S6(3N)S4
S27	249	S23 AND S26 AND S1
S28	102	S27 AND IC=G06F
S29	75	S28 NOT (S22 OR S19 OR S10)
S30	64	S29 NOT AD=20010714:20030714
S31	63	S30 NOT AD=20030714:20050701
S32	21	S31 AND IC=G06F-015
S33	4	S31 AND IC=G06F-007
S34	23	S32 OR S33
S35	23	IDPAT (sorted in duplicate/non-duplicate order)
S36	23	IDPAT (primary/non-duplicate records only)

File 347:JAPIO Nov 1976-2005/Feb(Updated 050606)
(c) 2005 JPO & JAPIO

File 350:Derwent WPIX 1963-2005/UD,UM &UP=200535
(c) 2005 Thomson Derwent

12/5/1 (Item 1 from file: 347)
DIALOG(R)File 347:JAPIO
(c) 2005 JPO & JAPIO. All rts. reserv.

00550009 **Image available**
TERMINAL EQUIPMENT

PUB. NO.: 55-037609 [JP 55037609 A]
PUBLISHED: March 15, 1980 (19800315)
INVENTOR(s): IMAI YASUHIRO
APPLICANT(s): HITACHI LTD [000510] (A Japanese Company or Corporation), JP
(Japan)
APPL. NO.: 53-108644 [JP 78108644]
FILED: September 06, 1978 (19780906)
INTL CLASS: [3] G06F-003/153 ; G09G-001/00
JAPIO CLASS: 44.9 (COMMUNICATION -- Other); 42.3 (ELECTRONICS -- Electron
Tubes); 45.3 (INFORMATION PROCESSING -- Input Output Units)
JOURNAL: Section: P, Section No. 12, Vol. 04, No. 70, Pg. 68, May 23,
1980 (19800523)

ABSTRACT

PURPOSE: To improve circuit efficiency with the reduced **data** transfer amount on a circuit, by sending out screen **data** of only a **change field**

CONSTITUTION: As for a CRT terminal equipment, respective lines on a screen are assigned to **fields** ; at the time of initial setting, screen buffer 4 is **cleared** and screen **data** in buffer 4 are CRC-computed 6, **field** by field, to **write** the results in CRC memory buffer 9 via register 7. On receiving a transmission request, the terminal equipment performs CRC arithmetic 6 of the contents of buffer 4, **field** (line) by **field** , in sequence from the 1st line to the **last** line **prior** to the editing of **data** to be sent. Then, it makes comparison 8 between the result obtained in register 7 and the result of **previous** CRC arithmetic 6 of a corresponding line (**field**) stored in buffer 9 and only when dissidence signal 14 is outputted, transmission control circuit 5 edits transmitted **data** about screen **data** of the corresponding **field** and sends it out to circuit 17.

12/5/2 (Item 1 from file: 350)
DIALOG(R)File 350:Derwent WPIX
(c) 2005 Thomson Derwent. All rts. reserv.

011109051 **Image available**
WPI Acc No: 1997-086976/199708
XRPX Acc No: N97-071728

Variant consistency evaluating method for hierarchical versioned data management system - involves providing for each variant drawdown-from identifier field for storing last - change identifier value of variant from which it was drawn down

Patent Assignee: INT BUSINESS MACHINES CORP (IBMC)
Inventor: EISENBERG N R; HUDDLESTON R L; KARASIUK G R; LEE T K; LEHNER M C;
TRAN B T; TRIBOLET C S

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
US 5592661	A	19970107	US 92914902	A	19920716	199708 B

Priority Applications (No Type Date): US 92914902 A 19920716

Patent Details:

Patent No	Kind	Lan Pg	Main IPC	Filing Notes
US 5592661	A		56 G06F-017/30	

Abstract (Basic): US 5592661 A

The method involves providing a **last - change identifier field** for identifying a most recent **change** to a variant. **Last - change identifier field** of the variant is **set** to a **value** currently **unused** for the part when the variant is created. The **last - change**

identifier field of the variant is **set** to a **value** currently **unused** for the part when the variant is updated.
A drawdown-from identifier **field** for storing the **last - change identifier value** of the variant from which it was drawn down is provided for each variant. The drawdown-from identifier **field** of the variant is **set** to the **value** of the **last - change identifier**

field of the variant from which it was drawn down.

ADVANTAGE - Has ability to detect independent **changes** to parts and control **data** integrity within variant hierarchy.

Dwg.15a/15

C

Title Terms: VARIANT; CONSISTENCY; EVALUATE; METHOD; HIERARCHY; **DATA** ;
MANAGEMENT; SYSTEM; VARIANT; DRAWDOWN; IDENTIFY; **FIELD** ; STORAGE; **LAST**
; **CHANGE** ; IDENTIFY; **VALUE** ; VARIANT; DRAW; DOWN

Derwent Class: T01

International Patent Class (Main): **G06F-017/30**

File Segment: EPI

21/5/4 (Item 4 from file: 347)
DIALOG(R)File 347:JAPIO
(c) 2005 JPO & JAPIO. All rts. reserv.

02332343 **Image available**
EXPANDING SYSTEM FOR CODASYL SYSTEM DATA BASE

PUB. NO.: 62-249243 [JP 62249243 A]
PUBLISHED: October 30, 1987 (19871030)
INVENTOR(s): OUCHI MASAHIRO
APPLICANT(s): MITSUBISHI ELECTRIC CORP [000601] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 61-094185 [JP 8694185]
FILED: April 23, 1986 (19860423)
INTL CLASS: [4] G06F-012/00
JAPIO CLASS: 45.2 (INFORMATION PROCESSING -- Memory Units)
JOURNAL: Section: P, Section No. 691, Vol. 12, No. 125, Pg. 64, April 19, 1988 (19880419)

ABSTRACT

PURPOSE: To reduce resources to be used, and to attain a fast reconstitution, by performing a reconstituting process in keeping with the **change** and the expansion of a **data base**, according to the physical structure of the **data base**.

CONSTITUTION: After the inputs of an **old data base** defining **field 2**, and a new **data base** defining **field 3**, a **change point** is checked, and a **change point table** which represents the **change point**, and a changing process at the **change point**, is generated. Next, a **data** in an **old data base 1** is inputted sequentially in the order of a physical structure, and it is decided whether the said data requires or not the **change** of the physical structure in referring to the changing table. When it is decided as necessary to **change**, the **change** of a structure is performed, and a new storing position, etc., for a **changed data** is decided, thereby an intermediate file 5 is completed. Hereinafter, the pointer of the data relating to the data in which a position is **changed**, etc., is **changed**, and the content of the intermediate file 5 is copied to a new **data base 7**.

21/5/5 (Item 1 from file: 350)
DIALOG(R) File 350:Derwent WPIX
(c) 2005 Thomson Derwent. All rts. reserv.

013013299 **Image available**
WPI Acc No: 2000-185150/200017
XRPX Acc No: N00-136772

Data processing apparatus for database updating function has type field indicating whether record is concrete record, or delta record that possibly modifies attribute value of a previous concrete or delta record

Patent Assignee: INT COMPUTERS LTD (INCM)
Inventor: GARDNER P A R
Number of Countries: 028 Number of Patents: 004
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
EP 984369	A2	20000308	EP 99305329	A	19990705	200017 B
JP 2000076105	A	20000314	JP 99237979	A	19990825	200024
CA 2278822	A1	20000229	CA 2278822	A	19990726	200033
US 6385616	B1	20020507	US 99354035	A	19990715	200235

Priority Applications (No Type Date): GB 9818819 A 19980829

Patent Details:

Patent No	Kind	Lan Pg	Main IPC	Filing Notes
EP 984369	A2	E 11	G06F-017/30	
Designated States (Regional): AL AT BE CH CY DE DK ES FI FR GB GR IE IT				
LI LT LU LV MC MK NL PT RO SE SI				
JP 2000076105	A		8 G06F-012/00	
CA 2278822	A1	E	G06F-017/30	
US 6385616	B1		G06F-017/30	

Abstract (Basic): EP 984369 A2

NOVELTY - The apparatus includes a device for storing records. Each record includes at least one **attribute**, a time span indicating for which the **attribute** is valid, an insertion time indicating when the record was created, and a type **field**, indicating whether the record is a concrete record, or a delta record that possibly **modifies** the **attribute value** of a **previous** concrete or delta record.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are included for:

(a) a data processing method
(b) a data carrier method including a computer program for performing the method

USE - In a time-versioned data storage mechanism.

ADVANTAGE - Supports the ability to view the state of the data from any selected point in time.

DESCRIPTION OF DRAWING(S) - The drawing is a flowchart showing a **database** update function.

pp; 11 DwgNo 2/5

Title Terms: DATA; PROCESS; APPARATUS; **DATABASE**; UPDATE; FUNCTION; TYPE; **FIELD**; INDICATE; RECORD; CONCRETE; RECORD; DELTA; RECORD; POSSIBILITY; **MODIFIED**; **ATTRIBUTE**; VALUE; CONCRETE; DELTA; RECORD

Derwent Class: T01

International Patent Class (Main): **G06F-012/00**; **G06F-017/30**

File Segment: EPI

21/5/6 (Item 2 from file: 350)
DIALOG(R)File 350:Derwent WPIX
(c) 2005 Thomson Derwent. All rts. reserv.

012408297 **Image available**
WPI Acc No: 1999-214405/199918
XRPX Acc No: N99-157808

Bitmapped index updating method in database system
Patent Assignee: ORACLE CORP (ORAC-N)
Inventor: COHEN J I; DEPLEDGE M; JAKOBSSON H; OZBUTUN C
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
US 5884307	A	19990316	US 97808585	A	19970228	199918 B

Priority Applications (No Type Date): US 97808585 A 19970228

Patent Details:

Patent No	Kind	Lan Pg	Main IPC	Filing Notes
US 5884307	A		18 G06F-017/30	

Abstract (Basic): US 5884307 A

NOVELTY - The sorted **change** records are applied to bitmap represented in the bitmapped index by locking bitmapped index entry containing a bitmap segment corresponding to the **change** made to data. The bitmap bit in the bitmapped index specified by the key value and **row** ID in the sorted **change** records is flipped, after which locked bitmapped index entry is unlocked.

DETAILED DESCRIPTION - Multiple **change** records reflecting **change** made to data are sorted based on key value and **row** ID which identify bitmap segment contained in bitmapped index to be updated. If the bitmapped index contains an entry with bitmap segment preceding the key and **row** number in **changed data**, then the **preceding** index entry is locked. If the bitmapped index does not contain an entry preceding key and **row** number to be **changed**, a dummy bitmapped index entry is created and stored in bitmapped index that is locked. INDEPENDENT CLAIMS are included for the following:

(a) a computer readable medium storing bitmapped index updating procedure;

(b) a computer system for updating bitmapped index.

USE - For **database** system.

ADVANTAGE - Minimizes lock resources required to update bitmapped indexes.

DESCRIPTION OF DRAWING(S) - The figure shows flow chart for illustrating bitmapped index updating method.

pp; 18 DwgNo 8/10

Title Terms: INDEX; UPDATE; METHOD; **DATABASE** ; SYSTEM

Derwent Class: T01

International Patent Class (Main): **G06F-017/30**

File Segment: EPI

22/5/1 (Item 1 from file: 350)
DIALOG(R)File 350:Derwent WPIX
(c) 2005 Thomson Derwent. All rts. reserv.

016665272 **Image available**
WPI Acc No: 2004-823992/200482
XRPX Acc No: N04-650774

Deleted row identification method in database management system,
involves scanning base table with delete marker that is set based on
identified last row in base table, and filtering results of scanning

Patent Assignee: MICROSOFT CORP (MICT)
Inventor: AGARWAL S H; ELLIS N R; KLINE R N; KRISHNAMOORTHY G; XIAO W
Number of Countries: 036 Number of Patents: 004
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
EP 1480132	A2	20041124	EP 200411092	A	20040510	200482 B
JP 2004348744	A	20041209	JP 2004152121	A	20040521	200482
US 20040236763	A1	20041125	US 2003445095	A	20030522	200482
KR 2004101077	A	20041202	KR 200436457	A	20040521	200525

Priority Applications (No Type Date): US 2003445095 A 20030522

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
EP 1480132	A2	E	21	G06F-017/30	
Designated States (Regional): AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IT LI LT LU LV MC MK NL PL PT RO SE SI SK TR					
JP 2004348744	A		21	G06F-012/00	
US 20040236763	A1			G06F-007/00	
KR 2004101077	A			G06F-017/40	

Abstract (Basic): EP 1480132 A2

NOVELTY - A last row in a base table that is **changed** during transaction **prior** to execution of a data manipulation language (DML) statement, is identified and a delete marker is set in response to the identified last row. The base table is scanned with the delete marker and the results of the scanning are filtered.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are also included for the following:

- (1) method for identifying rows inserted into base table;
- (2) system for identifying **rows deleted** from base table;
- (3) system for identifying rows inserted into base table;
- (4) computer readable medium storing **deleted row** identification program; and
- (5) computer readable medium storing inserted row identification program.

USE - For identifying **rows deleted** from base table during transaction in database management system (DBMS) used with personal computer (PC), hand-held device, server, router, peer device, multi-processor system, microprocessor-based or programmable consumer electronics, network PC, minicomputer, mainframe computer connected to local area network (LAN), wide area network (WAN), enterprise wide computer network, Internet or intranet.

ADVANTAGE - Enables identifying the **deleted rows** inexpensively, without inefficient use of disk input/output (I/O).

DESCRIPTION OF DRAWING(S) - The figure shows a block diagram of the database management system.

pp; 21 DwgNo 2/5

Title Terms: DELETE; ROW; IDENTIFY; METHOD; DATABASE; MANAGEMENT; SYSTEM;
SCAN; BASE; TABLE; DELETE; MARK; SET; BASED; IDENTIFY; LAST; ROW; BASE;
TABLE; FILTER; RESULT; SCAN

Derwent Class: T01

International Patent Class (Main): G06F-007/00; G06F-012/00; G06F-017/30;
G06F-017/40

File Segment: EPI

22/5/4 (Item 4 from file: 350)
DIALOG(R)File 350:Derwent WPIX
(c) 2005 Thomson Derwent. All rts. reserv.

013673761 **Image available**
WPI Acc No: 2001-157973/200116
XRPX Acc No: N01-114987

**Incremental refresh of materialized view in database management system,
by updating NULL values in selected columns to reflect new values and
inserts into selected table that occurred after prior refresh**

Patent Assignee: ORACLE CORP (ORAC-N)

Inventor: DIAS K; WITKOWSKI A

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
US 6134543	A	20001017	US 98109782	A	19980702	200116 B

Priority Applications (No Type Date): US 98109782 A 19980702

Patent Details:

Patent No	Kind	Lan Pg	Main IPC	Filing Notes
US 6134543	A	18	G06F-017/30	

Abstract (Basic): US 6134543 A

NOVELTY - A base table of materialized view is selected. If selected table is the right table of an outer join, selected columns of rows of materialized view are set to NULL based on rows of selected table that are updated or deleted after prior refresh. The NULL values in selected columns are updated to reflect new values caused by updates and inserts into selected table that occurred after prior refresh.

DETAILED DESCRIPTION - If the selected table is not the right table, then rows from materialized view are **deleted** based on **rows** of selected table that have been updated or deleted in the selected table after prior refresh operation. Then, rows are inserted into materialized view, based on updates and inserts into the selected table that occurred after prior refresh operation. INDEPENDENT CLAIMS are also included for the following:

- (a) computer readable medium;
- (b) database system

USE - For incremental refreshing of materialized view in database management system.

ADVANTAGE - Since incremental refresh techniques are memory less, requirement of record of sequence of **changes** to **prior** refresh is avoided.

DESCRIPTION OF DRAWING(S) - The figure shows the block diagram of two base tables at time T1 and materialized view defined by an outer join on the base tables.

pp; 18 DwgNo 4a, 4b/6

Title Terms: INCREMENT; REFRESH; VIEW; DATABASE; MANAGEMENT; SYSTEM; UPDATE
; NULL; VALUE; SELECT; COLUMN; REFLECT; NEW; VALUE; INSERT; SELECT; TABLE
; OCCUR; AFTER; PRIOR; REFRESH

Derwent Class: T01

International Patent Class (Main): G06F-017/30

File Segment: EPI

36/5/23 (Item 23 from file: 347)
DIALOG(R)File 347:JAPIO
(c) 2005 JPO & JAPIO. All rts. reserv.

04495325 **Image available**
METHOD FOR DISPLAYING SIMULATION RESULT

PUB. NO.: 06-139225 [JP 6139225 A]
PUBLISHED: May 20, 1994 (19940520)
INVENTOR(s): YOSHINO AKIYO
SADAI KEIJI
APPLICANT(s): FUJITSU LTD [000522] (A Japanese Company or Corporation), JP
(Japan)
APPL. NO.: 04-285934 [JP 92285934]
FILED: October 23, 1992 (19921023)
INTL CLASS: [5] **G06F-015/20** ; **G06F-015/24**
JAPIO CLASS: 45.4 (INFORMATION PROCESSING -- Computer Applications)
JAPIO KEYWORD: R131 (INFORMATION PROCESSING -- Microcomputers &
Microprocessors)
JOURNAL: Section: P, Section No. 1788, Vol. 18, No. 441, Pg. 148,
August 17, 1994 (19940817)

ABSTRACT

PURPOSE: To easily understand a difference in simulation result by displaying set data and simulation results before and after modification.

CONSTITUTION: Two simulation display **fields** before and after modification are displayed at a display part. When the **modification** of setting **data** is selected, data displayed in the left-side display **field** where **last** set **data** and a calculation result are displayed are copied to the right-side display **field**. Set **data** after the **modification** are inputted from an input part. The inputted data are recorded in the memory of a recording part and the set data displayed at the display part are modified. Then the modified part is meshed. Simulation is carried out by using the **modified** set **data**. The result of the simulation is recorded in the memory of the recording part and displayed at the display part.

Set	Items	Description
S1	6080165	ROW OR ROWS OR COLUMN? OR TUPLE? OR ATTRIBUTE? OR FIELD?
S2	8980393	REPOPULAT? OR RELOAD? OR REPLAC? OR REFILL? OR SWITCH? OR - MODIF? OR CHANGE? OR ALTER?
S3	3522471	FILL OR WRITE OR FILLING OR FILLS OR WRITING OR LOAD? OR - POPULATING OR POPULATES OR SET
S4	15673930	VALUE? OR DATA? OR RECORD? OR CELL OR CELLS
S5	132341	EMPTY OR UNFILLED OR UNUSED OR BLANK OR BLANKS OR UNPOPULA- TED OR CLEARED OR VACANT
S6	4043626	OLD OR PRIOR OR PREVIOUS OR LAST OR OLDER OR FORMER OR PAST OR PRECEDING OR EARLIER
S7	48	S1 AND S2 AND S3 AND S4 AND S5 AND S6
S8	969606	DATABASE? OR DATABANK? OR DATA() (BASE? OR BANK?) OR DB OR - OODB OR DBMS OR RDB OR DBMS OR DBS OR OODBS OR RDBS
S9	1679	S1(2N)S5
S10	36179	(S2 OR S3) AND S6(2N)S4
S11	0	S9 AND S10
S12	4698	S10 AND S1
S13	8	S9 AND S6(2N)S4
S14	20	S12 AND S5
S15	73	S7 OR S13 OR S14
S16	58	RD (unique items)
S17	47	S16 NOT PY>2001
S18	261	(REPOPULAT? OR RELOAD? OR REFILL? OR REPLAC?) (3N) (CELL OR - CELLS OR RECORD? OR VALUE?) (3N)S6
S19	15	S8 AND S18
S20	11	RD (unique items)
S21	11	S2 AND S3 AND S4 AND S5 AND S6 AND S8
S22	22	S20 OR S21
S23	16	RD (unique items)
S24	16	S23 NOT S17
S25	14	S24 NOT PY>2001
S26	29016	S8(2N) (DISTRIBUTED OR SHARED OR GROUPWARE? OR WORKGROUP? OR MULTI()USER)
S27	3784	S2 AND S26
S28	11061	S6(2N)S3
S29	4	S27 AND S28
S30	4	RD (unique items)
S31	2	S30 NOT PY>2001
S32	1	S27 AND S6 AND S4 AND S5
File	8: Ei Compendex(R) 1970-2005/May W5	
	(c) 2005 Elsevier Eng. Info. Inc.	
File	35: Dissertation Abs Online 1861-2005/May	
	(c) 2005 ProQuest Info&Learning	
File	65: Inside Conferences 1993-2005/Jun W1	
	(c) 2005 BLDSC all rts. reserv.	
File	2: INSPEC 1969-2005/May W5	
	(c) 2005 Institution of Electrical Engineers	
File	94: JICST-EPlus 1985-2005/Apr W3	
	(c) 2005 Japan Science and Tech Corp (JST)	
File	111: TGG Natl. Newspaper Index (SM) 1979-2005/Jun 06	
	(c) 2005 The Gale Group	
File	6: NTIS 1964-2005/May W5	
	(c) 2005 NTIS, Intl Cpyrghrt All Rights Res	
File	144: Pascal 1973-2005/May W5	
	(c) 2005 INIST/CNRS	
File	434: SciSearch(R) Cited Ref Sci 1974-1989/Dec	
	(c) 1998 Inst for Sci Info	
File	34: SciSearch(R) Cited Ref Sci 1990-2005/May W5	
	(c) 2005 Inst for Sci Info	
File	99: Wilson Appl. Sci & Tech Abs 1983-2005/May	
	(c) 2005 The HW Wilson Co.	
File	95: TEME-Technology & Management 1989-2005/May W1	
	(c) 2005 FIZ TECHNIK	

25/5/4 (Item 4 from file: 8)
DIALOG(R)File 8: Ei Compendex(R)
(c) 2005 Elsevier Eng. Info. Inc. All rts. reserv.

03701497 E.I. No: EIP93091069213

Title: Simple analysis of the LRU buffer policy and its relationship to buffer warm-up transient

Author: Bhide, Anupam K.; Dan, Asit; Dias, Daniel M.

Corporate Source: IBM Thomas J. Watson Research Cent, Yorktown Heights, NY, USA

Conference Title: 1993 IEEE 9th International Conference on Data Engineering

Conference Location: Vienna, Austria Conference Date: 19930419-19930423

Sponsor: IEEE Computer Society; Technical Committee on Data Engineering; Austrian Computer Society (OCG)

E.I. Conference No.: 18785

Source: 1993 IEEE 9th International Conference on Data Engineering Proceedings - International Conference on Data Engineering 1993. Publ by IEEE, Computer Society Press, Los Alamitos, CA, USA. p 125-131

Publication Year: 1993

CODEN: PIDEEG ISBN: 0-8186-3570-3

Language: English

Document Type: CA; (Conference Article) Treatment: A; (Applications); G; (General Review); T; (Theoretical)

Journal Announcement: 9310W5

Abstract: In a **database** system, a buffer hit transient may occur after a **load** surge or a node failure. The buffer warm-up transient is the expected buffer hit probability as a function of the time, starting with an **empty** buffer until the buffer is full. It is only after the buffer becomes full that a buffer **replacement** policy kicks in. The buffer transient analysis presented in this paper has three main contributions. First, we present a simple approximation for the buffer warm-up transient, and show that it agrees very well with simulation estimates. This result can be used to estimate how long it takes for the buffer hit probability to get to any specified fraction of its steady state **value**, starting with an **empty** buffer, say after a failure. Secondly, we show that the analysis for the buffer warm-up transient leads to a simple and very accurate estimate of the steady state buffer hit probability for the LRU buffer **replacement** policy. **Previous** approximations to the LRU policy are comparatively more complicated, but we show that they result in estimates indistinguishable from the simple analysis we present. Thirdly, we generalize this method to estimate the transient behavior of the LRU policy starting with a non-**empty** buffer. This method can be used, for instance, to estimate the effect of a **load** surge on the buffer hit probability. We show that after a short **load** surge, it can take much longer than the surge duration for the buffer hit probability to return to its steady state **value**. (Author abstract) 15 Refs.

Descriptors: *Database systems; Computer system recovery; Random access storage; Information retrieval; **Data** transfer; Probability; Estimation; Algorithms; Computer simulation

Identifiers: Least recently used (LRU) buffer **replacement** policy; Buffer warm up transients; Buffer hit probability; Node failure; **Load** surge

Classification Codes:

723.3 (Database Systems); 721.1 (Computer Theory, Includes Formal Logic, Automata Theory, Switching Theory, Programming Theory); 722.1 (Data Storage, Equipment & Techniques); 903.3 (Information Retrieval & Use); 723.2 (Data Processing); 922.1 (Probability Theory)

723 (Computer Software); 721 (Computer Circuits & Logic Elements); 722 (Computer Hardware); 903 (Information Science); 922 (Statistical Methods)

72 (COMPUTERS & DATA PROCESSING); 90 (GENERAL ENGINEERING); 92 (ENGINEERING MATHEMATICS)

25/5/7 (Item 3 from file: 2)
DIALOG(R)File 2:INSPEC
(c) 2005 Institution of Electrical Engineers. All rts. reserv.

4535174 INSPEC Abstract Number: B9401-6150J-018, C9401-6120-028
Title: Teletraffic theory applied to the analysis of hash-structured files
Author(s): Cooper, R.B.; Solomon, M.K.
Author Affiliation: Dept. of Comput. Sci., Florida Atlantic Univ., Boca Raton, FL, USA
Journal: Archiv fur Elektronik und Uebertragungstechnik vol.47, no.5-6 p.336-41

Publication Date: Sept.-Nov. 1993 Country of Publication: West Germany
CODEN: AEUTAH ISSN: 0001-1096
Language: English Document Type: Journal Paper (JP)
Treatment: Theoretical (T)
Abstract: Certain hash-structured files consist of sequences (chains) of computer memory locations (slots) into which records are inserted, and from which they are later retrieved or deleted. Assuming that the records arrive to the file according to a Poisson process for insertion into a chain (randomly selected by the hash function), and reside in memory for a random length of time before being deleted, then this can be associated with a teletraffic model in which the records are "calls" and the slots in the chain are "trunks." In particular, the authors consider two models: (1) hardpack, where a deleted record is immediately physically removed from the file and is **replaced** by the **last record** on its chain, and (2) softpack, where a deleted record is marked, and a subsequent arriving record is written over the first marked record on its chain. The slots can be accessed individually or in groups (called "buckets"). In each case the authors are interested in the length of a successful search (the number of buckets inspected until an arbitrary record is located) and the length of an unsuccessful search (the number of buckets inspected until it is determined that an arbitrary record is not in the file). If they identify these file structures as analogous to trunk groups with ordered hunt, then these models can be analyzed using results obtained many years ago (in particular, Kosten (1937) and Burke (1971)) for teletraffic applications, results which are not well known or easily accessible to the **database**-performance-analysis community. (10 Refs)

Subfile: B C B C
Descriptors: **database** theory; file organisation; queueing theory; telecommunication traffic
Identifiers: teletraffic theory; computer memory locations; Poisson process; hardpack model; softpack model; buckets; successful search length; unsuccessful search length; trunk groups; ordered hunt; **database**-performance-analysis; hash-structured files; queueing theory
Class Codes: B6150J (Queueing systems); C6120 (File organisation); B0240C (Queueing theory); C4250 (Database

25/5/10 (Item 6 from file: 2)
DIALOG(R)File 2:INSPEC
(c) 2005 Institution of Electrical Engineers. All rts. reserv.

02681548 INSPEC Abstract Number: C86035010

Title: A look at self-maintaining database area logic. II (shortcomings of using the 'WITHIN' clause to control physical record placement)

Author(s): Hill, J.

Journal: Unisphere vol.5, no.12 p.35-8

Publication Date: March 1986 Country of Publication: USA

CODEN: UNISD2 ISSN: 0279-1579

Language: English Document Type: Journal Paper (JP)

Treatment: Practical (P)

Abstract: For pt.I see ibid., vol.V, no.11, p.14-16 (1986). One of the significant problems that stems from using the WITHIN clause, and the resultant forcing of the ILLOGICAL INSERT POINT as the primary physical placement mechanism, may not become apparent until the **database** has acquired some 'age' and the area is filled. Veteran **database** administrators have faced this problem for several years, and many have elected to write UNLOAD/RELOAD utilities to repack the records stored in the area and to force, under INITIAL LOAD conditions, the ILLOGICAL INSERT POINT to reset to at least the **last record** stored (**RELOADED**). The two Area **Record** Placement mechanisms supported by DMS/1100 (DARP and SARP) have evolved and changed in an attempt to manage such an overflow of records as would occur in our example if we allowed the WITHIN clause to control physical placement. The solution to the record placement dilemma should be as simple as it is obvious: restore the use of the LOGICAL INSERT POINT in areas where the owner records are stored external to the Area of the dependent records. (0 Refs)

Subfile: C

Descriptors: **database** management systems; storage management

Identifiers: physical record placement; Area Record Placement mechanisms; DMS/1100; DARP; SARP; LOGICAL INSERT POINT

25/5/14 (Item 1 from file: 95)
DIALOG(R)File 95:TEME-Technology & Management
(c) 2005 FIZ TECHNIK. All rts. reserv.

00697944 E93083370089

Complex objects in the temporal object system

(Komplexe Objekte in einem zeitlichen Objektsystem)

Fotouhi, F; Shah, AA; Grosky, W

Wayne State Univ., Detroit, USA

ICCI '92, 4th Int. Conf. on Computing and Inf., Toronto, CDN, May 28-30, 19921992

Document type: Conference paper Language: English

Record type: Abstract

ISBN: 0-8186-2812-X

ABSTRACT:

In existing relational **database** systems, data objects are stored in a non-temporal fashion. That is, when the **value** of an attribute changes, the **old value** is **replaced** by the new **value**. Thus, only the latest state of an object resides in the **database**. However, for many **database** applications such as CAD/CAM, it is not appropriate to discard old information. In these cases it is necessary to associate time values with data to indicate the time for which the data is valid. In many engineering applications, changes to the state and/or structure of an object needs to be maintained over a period of time Existing object-oriented data models allow such changes in the state (referred to as Version Management) and structure (referred to as Schema Evolution) of an object. However, when the structure changes, the old structure is replaced by the new one. In this paper propose a Temporal Object System (TOS) which maintains changes to both the structure and the state of an object. Objects in this system are referred to Temporal Objects and are allowed to evolve overtime. In this paper discuss how to extend TOS in order to construct Complex Temporal Objects from an aggregation of temporal objects.

DESCRIPTORS: RELATIONAL **DATABASES** ; COMPLEX SYSTEM; **DATABASE** MANAGEMENT SYSTEM; TIME BEHAVIOUR; DATA ORGANIZATION; DATA MODELS
IDENTIFIERS: OBJEKT; Relationsdatenbank; Datenmodell

31/5/2 (Item 2 from file: 35)
DIALOG(R)File 35:Dissertation Abs Online
(c) 2005 ProQuest Info&Learning. All rts. reserv.

899043 ORDER NO: AAD85-24396
**UPDATE SYNCHRONIZATION PROBLEM IN DISTRIBUTED CONTROL DATABASE SYSTEM
(NETWORK, CONCURRENCY)**
Author: OMAR, KHAMIS ALI
Degree: PH.D.
Year: 1985
Corporate Source/Institution: SOUTHERN METHODIST UNIVERSITY (0210)
Source: VOLUME 46/09-B OF DISSERTATION ABSTRACTS INTERNATIONAL.
PAGE 3126. 209 PAGES
Descriptors: COMPUTER SCIENCE
Descriptor Codes: 0984

A class of **distributed databases**, called **distributed control database** (DCDB) has been identified. The DCDB is a set of data items stored and maintained about the status and the behavior of the network itself. The network could be packet (ARPANET), circuit **switching** (voice or data), or local network. Decisions are continuously taken in each node, depending on the contents of database, to route the commodity (calls, data, planes,....) through the network.

DCDBs have been shown to exhibit special properties which make them different from the traditional **distributed database** systems. Their transactions are mainly read-only or write-only (write without knowing current value of the data). Old data versions are acceptable, in certain cases, to satisfy the system queries. We investigate the idea of customizing algorithms to this particular model and show that it can lead to simpler, more efficient algorithms. Utilizing the DCDB special features, an update synchronization method, GUSP, has been developed. The GUSP gives high priority to read-only transactions and synchronizes the updates so that all the updates of a logical data item are managed by one node. Since the updates are **write** -only, **old** updates are discarded and their results are assumed to be overwritten by the more recent ones. Mutual data consistency is relaxed so that accessing of the database is allowable during the synchronization process. The performance of the GUSP has been evaluated, using simulation method, and compared with other existing protocols.